

What is claimed is:

1 *Sub A* 1. A program converting unit for generating a machine
2 language instruction from a source program for a processor
3 that manages an N-bit address while processing M-bit data, N
4 being greater than M, said program converting unit comprising:
5 parameter holding means for holding a data width and a
6 pointer width designated by a user, said data width
7 representing the number of bits of data used in the source
8 program while said pointer width representing the number of
9 bits of an address; and
10 generating means for generating an instruction to manage
11 said data width when a variable operated by said instruction
12 represents the data, and for generating an instruction to
13 manage said pointer width when a variable operated by said
14 instruction represents the address.

1 2. The program converting unit of Claim 1, wherein
2 said M is 16 and said N is an integer in a range of 17 to 31
3 inclusive.

1 3. The program converting unit of Claim 1, wherein
2 said generating means includes:
3 *determining* *determining*
judging means for *judging* a kind of the machine language
4 instruction, the machine language instruction including (1) an
5 instruction to access to a memory, (2) an instruction to use
6 a register, and (3) an instruction to use an immediate;

7 memory managing means for outputting a direction, in
8 case of the (1) instruction, to manage said data width as an
9 effective memory-access width when a variable to be accessed
10 represents the data, and to manage said pointer width as an
11 effective memory-access width when said variable represents the
12 address;

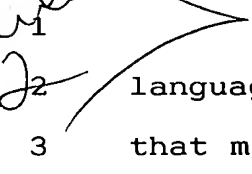
13 register managing means for outputting a direction, in
14 case of the (2) instruction, to manage said data width as an
15 effective bit-width when a variable to be read/written
16 from/into the register represents the data, and to manage said
17 pointer width as the effective bit-width when said variable
18 represents the address;

19 immediate managing means for outputting a direction,
20 in case of the (3) instruction, to manage said data width as
21 the effective bit-width when said immediate represents the
22 data, and to manage said pointer width as the effective bit-
23 width when said immediate represents the address; and

24 code generating means for generating the machine
25 language instruction in accordance with the directions from
26 said memory managing means, said register managing means, and
27 said immediate managing means.

1 4. The program converting unit of Claim 3, wherein
2 said M is 16 and said N is an integer in a range of 17 to 31
3 inclusive.

1 5. The program converting unit of Claim 4, wherein:
2 said N is 24; and
3 said code generating means generates an instruction for
4 a 24-bit data operation when said pointer width is greater than
5 16 bits and less than 24 bits, and generates an instruction for
6 a 16-bit data operation when said pointer width is 16 bits or
7 less.


1 6. A program converting unit for generating a machine
2 language instruction based on a source program for a processor
3 that manages an N-bit address while processing M-bit data, N
4 being greater than M, said program converting unit comprising:
5 syntax analyzing means for analyzing a syntax of the
6 source program to convert the same into an intermediary
7 language comprising intermediary instructions, and subsequently
8 for judging whether or not each variable contained in said
9 intermediary instructions represents data used in an address;
10 table generating means for generating a table for each
11 variable in said intermediary instructions, said table holding
12 a name together with a type of each variable, said type
13 representing one of the data and the address;
14 parameter holding means for holding a data width and a
15 pointer width designated by a user, said data width
16 representing the number of bits of the data while said pointer
17 width representing the number of bits of the address; and
18 generating means for generating an instruction to manage

19 said data width when the variable in said intermediary
20 instruction represents the data, and an instruction to manage
21 said pointer width when said variable represents the address.

1 7. The program converting unit of Claim 6, wherein
2 said M is 16 and said N is an integer in a range of 17 to 31
3 inclusive.

1 8. The program converting unit of Claim 6, wherein said
2 generating means includes:

3 judging means for judging a kind of the machine language
4 instruction, the machine language instruction including (1) an
5 instruction to access to an memory, (2) an instruction to use
6 a register, and (3) an instruction to use an immediate;

7 memory managing means for outputting a direction, in
8 case of the (1) instruction, to manage a corresponding bit-
9 width held in said parameter holding means as an effective
10 memory-access width depending on the type of a variable to be
11 accessed shown in said table;

12 register managing means for outputting a direction, in
13 case of the (2) instruction, to manage a corresponding bit-
14 width held in said parameter holding means as an effective bit-
15 width depending on the type of a variable to be read/written
16 from/in the register shown in said table;

17 immediate managing means for outputting a direction, in
18 case of the (3) instruction, to manage a corresponding bit-

19 width held in said parameter holding means for the immediate
20 as an effective bit-width depending on the type of the
21 immediate shown in said table; and
22 code generating means for generating the machine
23 language instruction in accordance with the directions from
24 said memory managing means, said register managing means, and
25 said immediate managing means.

1 9. The program converting unit of Claim 8,
2 said M is 16 and said N is an integer in a range of 17 to 31
3 inclusive.

1 10. The program converting unit of Claim 9, wherein:
2 said N is 24; and
3 said code generating means generates an instruction for
4 a 24-bit data operation when said pointer width is greater than
5 16 bits and less than 24 bits, and generates an instruction for
6 a 16-bit data operation when said pointer width is 16 bits or
7 less.

1 11. A program converting unit for generating a machine
2 language instruction based on a source program for a processor
3 that manages an N-bit address while processing M-bit data, N
4 being greater than M, said program converting unit comprising:
5 syntax analyzing means for analyzing a syntax of the
6 source program to convert the same into an intermediary

7 language comprising intermediary instructions, and subsequently
8 for judging whether or not each variable contained in said
9 intermediary instructions represents data used in an address;
10 table generating means for generating a table for each
11 variable in said intermediary instructions, said table holding
12 a name together with a type of each variable, said type
13 representing one of the data and the address;
14 parameter holding means for holding a data width and a
15 pointer width designated by a user, said data width
16 representing the number of bits of the data while said pointer
17 width representing the number of bits of the address;
18 judging means for judging a kind of the machine language
19 instruction, the machine language instruction including (1) an
20 instruction to access to an memory, (2) an instruction to use
21 a register, and (3) an instruction to use an immediate;
22 memory managing means for outputting a direction, in
23 case of the (1) instruction, to manage a corresponding bit-
24 width held in said parameter holding means as an effective
25 memory-access width depending on the type of a variable to be
26 accessed shown in said table;
27 register managing means for outputting a direction, in
28 case of the (2) instruction, to manage a corresponding bit-
29 width held in said parameter holding means as an effective bit-
30 width depending on the type of a variable to be read/written
31 from/in the register shown in said table;
32 immediate managing means for outputting a direction, in

33 case of the (3) instruction, to manage a corresponding bit-
34 width held in said parameter holding means for the immediate
35 as an effective bit-width depending on the type of the
36 immediate shown in said table; and

37 code generating means for generating the machine
38 language instruction in accordance with the directions from
39 said memory managing means, said register managing means, and
40 said immediate managing means.

1 12. The program converting unit of Claim 11, wherein
2 said code generating means generates an instruction for
3 a 24-bit data operation when said pointer width is greater than
4 16 bits and less than 24 bits, and generates an instruction for
5 a 16-bit data operation when said pointer width is 16 bits or
6 less.

1 *Sub No* 13. A program converting unit for generating a machine
2 language instruction from a source program for a processor that
3 manages an N-bit address while processing M-bit data, N being
4 greater than M, said program converting unit comprising:

5 parameter holding means for holding a data width and a
6 pointer width designated by a user, said data width
7 representing the number of bits of data used in the source
8 program while said pointer width representing the number of
9 bits of an address;

10 generating means for generating an instruction to manage

11 said data width when a variable operated by said instruction
12 represents the data, and for generating an instruction to
13 manage said pointer width when a variable operated by said
14 instruction represents the address;

15 option directing means for holding a user's direction
16 for an overflow compensation, an overflow being possibly caused
17 by an arithmetic operation; and

18 compensate instruction generating means for generating
19 a compensation instruction to compensate an overflow in
20 accordance with a type of a variable used in the arithmetic
21 operation, said type being judged when said option directing
22 means holds the user's direction for executing the overflow
23 compensation, said compensation instruction being generated
24 when an effective bit-width of a variable designated by an
25 operand is shorter than a register of N-bit wide and the
26 arithmetic operation instruction will possibly cause an
27 overflow exceeding said effective bit-width.

1 14. The program converting unit of Claim 13, wherein
2 said M is 16 and said N is an integer in a range of 17 to 31
3 inclusive.

1 15. The program converting unit of Claim 13, wherein
2 said M is 32, and said N is an integer in a range of 33 to 63
3 inclusive.

1 16. The program converting unit of Claim 13, wherein
2 said compensate instruction generating means includes:
3 instruction judging means for judging an arithmetic
4 operation instruction that will possibly cause an overflow for
5 all the machine language instructions when said option
6 instructing means holds the user's direction for executing the
7 overflow compensation;
8 variable judging means, with respect to a variable in
9 the arithmetic operation instruction judged by said instruction
10 judging means, for judging an effective bit-width and whether
11 said variable is signed or unsigned by referring to said table;
12 sign-extension instruction generating means for
13 generating a compensation instruction in case of a signed
14 variable, a logical value of a sign bit being filled into all
15 bits higher than the effective bit-width in a register that
16 is to store said signed variable by said sign-extension
17 compensation instruction; and
18 zero-extension instruction generating means for
19 generating a zero-extension compensation instruction in case
20 of an unsigned variable, a logical value "0" being filled into
21 all bits higher than the effective bit width in a register
22 that is to store said unsigned variable by said zero-extension
23 compensation instruction.

1 17. The program converting unit of Claim 16, wherein
2 said generating means includes:

3 ^{determining} ~~judging~~ means for ^{determining} ~~judging~~ a kind of the machine language
4 instruction, the machine language instruction including (1) an
5 instruction to access to a memory, (2) an instruction to use
6 a register, and (3) an instruction to use an immediate;
7 memory managing means for outputting a direction, in
8 case of the (1) instruction, to manage said data width as an
9 effective memory-access width when a variable to be accessed
10 represents the data, and to manage said pointer width as an
11 effective memory-access width when said variable represents the
12 address;
13 register managing means for outputting a direction, in
14 case of the (2) instruction, to manage said data width as an
15 effective bit-width when a variable to be read/written
16 from/into the register represents the data, and to manage said
17 pointer width as the effective bit-width when said variable
18 represents the address;
19 immediate managing means for outputting a direction,
20 in case of the (3) instruction, to manage said data width as
21 the effective bit-width when said immediate represents the
22 data, and to manage said pointer width as the effective bit-
23 width when said immediate represents the address; and
24 code generating means for generating the machine
25 language instruction in accordance with the directions from
26 said memory managing means, said register managing means, and
27 said immediate managing means.

1 18. The program converting unit of Claim 17, wherein
2 said M is 16 and said N is an integer in a range of 17 to 31
3 inclusive.

1 19. The program converting unit of Claim 17, wherein
2 said M is 32, and said N is an integer in a range of 33 to 63
3 inclusive.

1 *Self AB* 20. A program converting unit for generating a machine
2 language instruction based on a source program for a processor
3 that manages an N-bit address while processing M-bit data, N
4 being greater than M, said program converting unit comprising:
5 syntax analyzing means for analyzing a syntax of the
6 source program to convert the same into an intermediary
7 language comprising intermediary instructions, and subsequently
8 for judging whether or not each variable contained in said
9 intermediary instructions represents data used in an address;
10 table generating means for generating a table for each
11 variable in said intermediary instructions, said table holding
12 a name together with a type of each variable, said type
13 representing one of the data and the address, and one of signed
14 and unsigned data;
15 parameter holding means for holding a data width and a
16 pointer width designated by a user, said data width
17 representing the number of bits of the data while said pointer
18 width representing the number of bits of the address;

19 option directing means for holding a user's direction
20 for an overflow compensation, an overflow being possibly caused
21 by an arithmetic operation;

22 generating means for generating an instruction to manage
23 said data width when the variable in said intermediary
24 instruction represents the data, and an instruction to manage
25 said pointer width when said variable represents the address;
26 and

27 compensate instruction generating means for generating
28 a compensation instruction to compensate an overflow in
29 accordance with a type of a variable used in the arithmetic
30 operation, said type being judged when said option directing
31 means holds the user's direction for executing the overflow
32 compensation, said compensation instruction being generated
33 when an effective bit-width of a variable designated by an
34 operand is shorter than a register of N-bit wide and the
35 arithmetic operation instruction will possibly cause an
36 overflow exceeding said effective bit-width.

1 21. The program converting unit of Claim 20, wherein
2 said M is 16 and said N is an integer in a range of 17 to 31
3 inclusive.

1 24
2 22. The program converting unit of Claim 20, wherein
3 said M is 32, and said N is an integer in a range of 33 to 63
4 inclusive.

25
23. The program converting unit of Claim 20, wherein
said compensate instruction generating means includes:
instruction judging means for judging an arithmetic
operation instruction that will possibly cause an overflow for
all the machine language instructions when said option
instructing means holds the user's direction for executing the
overflow compensation;
variable judging means, with respect to a variable in
the arithmetic operation instruction judged by said instruction
judging means, for judging an effective bit-width and whether
said variable is signed or unsigned by referring to said table;
sign-extension instruction generating means for
generating a compensation instruction in case of a signed
variable, a logical value of a sign bit being filled into all
bits higher than the effective bit-width in a register that
is to store said signed variable by said sign-extension
compensation instruction; and
zero-extension instruction generating means for
generating a zero-extension compensation instruction in case
of an unsigned variable, a logical value "0" being filled into
all bits higher than the effective bit width in a register
that is to store said unsigned variable by said zero-extension
compensation instruction.

26
24. The program converting unit of Claim 23, wherein
25

2 said generating means includes: .
3 ^{determining} ~~judging~~ means for ^{determining} ~~judging~~ a kind of the machine language
4 instruction, the machine language instruction including (1) an
5 instruction to access to an memory, (2) an instruction to use
6 a register, and (3) an instruction to use an immediate;
7 memory managing means for outputting a direction, in
8 case of the (1) instruction, to manage a corresponding bit-
9 width held in said parameter holding means as an effective
10 memory-access width depending on the type of a variable to be
11 accessed shown in said table;
12 register managing means for outputting a direction, in
13 case of the (2) instruction, to manage a corresponding bit-
14 width held in said parameter holding means as an effective bit-
15 width depending on the type of a variable to be read/written
16 from/in the register shown in said table;
17 immediate managing means for outputting a direction, in
18 case of the (3) instruction, to manage a corresponding bit-
19 width held in said parameter holding means for the immediate
20 as an effective bit-width depending on the type of the
21 immediate shown in said table; and
22 code generating means for generating the machine
23 language instruction in accordance with the directions from
24 said memory managing means, said register managing means, and
25 said immediate managing means.

22
1 25. The program converting unit of Claim 21, wherein
2 said M is 16 and said N is an integer in a range of 17 to 31
3 inclusive.

23
1 26. The program converting unit of Claim 21, wherein
2 said M is 32, and said N is an integer in a range of 33 to 63
3 inclusive.

Sub A4
1 27. A program converting unit for generating a machine
2 language instruction based on a source program for a processor
3 that manages an N-bit address while processing M-bit data, N
4 being greater than M, said program converting unit comprising:
5 syntax analyzing means for analyzing a syntax of the
6 source program to convert the same into an intermediary
7 language comprising intermediary instructions, and subsequently
8 for judging whether or not each variable contained in said
9 intermediary instructions represents data used in an address;
10 table generating means for generating a table for each
11 variable in said intermediary instructions, said table holding
12 a name together with a type of each variable, said type
13 representing one of the data and the address, and one of signed
14 and unsigned data;
15 parameter holding means for holding a data width and a
16 pointer width designated by a user, said data width
17 representing the number of bits of the data while said pointer
18 width representing the number of bits of the address;

19 option directing means for holding a user's direction
20 for an overflow compensation, an overflow being possibly caused
21 by an arithmetic operation;

22 generating means for generating an instruction to manage
23 said data width when the variable in said intermediary
24 instruction represents the data, and an instruction to manage
25 said pointer width when said variable represents the address;
26 and

27 compensate instruction generating means for generating
28 a compensation instruction to compensate an overflow in
29 accordance with a type of a variable used in the arithmetic
30 operation, said type being judged when said option directing
31 means holds the user's direction for executing the overflow
32 compensation, said compensation instruction being generated
33 when an effective bit-width of a variable designated by an
34 operand is shorter than a register of N-bit wide and the
35 arithmetic operation instruction will possibly cause an
36 overflow exceeding said effective bit-width, wherein
37 said generating means includes:

38 judging means for judging a kind of the machine language
39 instruction, the machine language instruction including (1) an
40 instruction to access to an memory, (2) an instruction to use
41 a register, and (3) an instruction to use an immediate;

42 memory managing means for outputting a direction, in
43 case of the (1) instruction, to manage a corresponding bit-
44 width held in said parameter holding means as an effective

45 memory-access width depending on the type of a variable to be
46 accessed shown in said table;

47 register managing means for outputting a direction, in
48 case of the (2) instruction, to manage a corresponding bit-
49 width held in said parameter holding means as an effective bit-
50 width depending on the type of a variable to be read/written
51 from/in the register shown in said table;

52 immediate managing means for outputting a direction, in
53 case of the (3) instruction, to manage a corresponding bit-
54 width held in said parameter holding means for the immediate
55 as an effective bit-width depending on the type of the
56 immediate shown in said table; and

57 code generating means for generating the machine
58 language instruction in accordance with the directions from
59 said memory managing means, said register managing means, and
60 said immediate managing means, and wherein

61 said compensate instruction generating means includes:

62 instruction judging means for judging an arithmetic
63 operation instruction that will possibly cause an overflow for
64 all the machine language instructions when said option
65 instructing means holds the user's direction for executing the
66 overflow compensation;

67 variable judging means, with respect to a variable in
68 the arithmetic operation instruction judged by said instruction
69 judging means, for judging an effective bit-width and whether
70 said variable is signed or unsigned by referring to said table;

71 sign-extension instruction generating means for
72 generating a compensation instruction in case of a signed
73 variable, a logical value of a sign bit being filled into all
74 bits higher than the effective bit-width in a register that
75 is to store said signed variable by said sign-extension
76 compensation instruction; and

77 zero-extension instruction generating means for
78 generating a zero-extension compensation instruction in case
79 of an unsigned variable, a logical value "0" being filled into
80 all bits higher than the effective bit width in a register that
81 is to store said unsigned variable by said zero-extension
82 compensation instruction.

1 28. A processor improved in address management
2 comprising:

3 memory means for storing a program including an N-bit
4 data arithmetic operation instruction and both N-bit and M-
5 bit data load/store instructions, N being greater than M;

6 a program counter for holding an N-bit instruction
7 address to output the same to said memory means;

8 fetching means for fetching an instruction from said
9 memory means using the instruction address from said program
10 counter; and

11 executing means for executing all N-bit arithmetic
12 operation instructions and for executing N-bit and M-bit
13 instructions excluding the arithmetic operation instructions,

14 whereby an N-bit address is calculated by the N-bit
15 arithmetic operation independently of a data bit-width, said
16 data bit-width being M.

1 29. The processor of Claim 28, further comprising:
2 an address register group including a plurality of N-
3 bit address registers;
4 a data register group including a plurality of N-bit
5 data registers,
6 wherein said executing means executes the N-bit and M-
7 bit data operation instructions using the address registers,
8 while executing the M-bit data operation instruction using the
9 data registers.

1 30. The processor of Claim 29, wherein :
2 said N is 24 and said M is 16; and
3 said processor is installed in a 1-chip microcomputer,
4 whereby said 1-chip microcomputer becomes suitable for
5 running a program that utilizes a memory over 64 Kbyte for an
6 operation with 16-bit data.

1 31. The processor of Claim 30, further comprising:
2 compensating means for extending an effective bit-width
3 of the data in one of the address register and the data
4 register to 24 bits,
5 wherein said compensating means operates in accordance

6 with said compensate instruction entered ~~immediately~~ after a
7 machine language instruction designating an arithmetic
8 operation that will possibly cause an overflow.

1 32. The processor of Claim 31, wherein said
2 compensating means includes:

3 a first extending unit for filling a logical value of
4 a sign bit in all bits higher than the effective bit-width in
5 a register;

6 a second extending unit for filling a logical value "0"
7 in all bits higher than the effective bit-width in a register.

1 33. The processor of Claim 28, further comprising:

2 an address register group including a plurality of N-
3 bit address registers; and

4 a data register group including a plurality of M-bit
5 data registers,

6 wherein said executing means executes one of an N-bit
7 data operation instruction and an M-bit data operation
8 instruction using the address registers, while executing the
9 M-bit data operation instruction using the data registers. .

1 34. The processor of Claim 33, wherein :

2 said N is 24 and said M is 16; and

3 said processor is installed in a 1-chip microcomputer,

4 whereby said 1-chip microcomputer becomes suitable for

5 running a program that utilizes a memory over 64 Kbyte for an
6 operation with 16-bit data.

1 35. The processor of Claim 34, further comprising:
2 compensating means for extending an effective bit-width
3 of the data in one of the address register and the data
4 register to 24 bits,

5 wherein said compensating means operates in accordance
6 with said compensate instruction entered immediately after a
7 machine language instruction designating an arithmetic
8 operation that will possibly cause an overflow.

1 36. The processor of Claim 35, wherein said
2 compensating means includes:

3 a first extending unit for filling a logical value of
4 a sign bit in all bits higher than the effective bit-width in
5 a register;

6 a second extending unit for filling a logical value "0"
7 in all bits higher than the effective bit-width in a register.

28
1 37. A processor for processing data in accordance with
2 instructions in a program comprising:

3 register means including a plurality of register groups,
4 each group being identical in bit-width while being different
5 in types;

6 instruction decoding means for decoding an instruction

7 to output register information indicating a register designated
8 by an operand contained in a data-transfer instruction;

9 external-access-width control means for outputting the
10 number of effective bits as bit-width information indicating
11 a bit-width of transmission data in accordance with a kind of
12 a register group to which said designated register belongs; and

13 external-access executing means for executing data
14 transfer between said designated register and an external
15 memory in accordance with said register information and said
16 bit-width information.

29 28
1 ~~38~~. The processor of Claim ~~37~~, wherein
2 said register means includes:

3 an address register group including a plurality of
4 address registers holding addresses; and

5 a data register group including a plurality of data
6 registers holding data.

30 29
1 ~~39~~. The processor of Claim ~~38~~, wherein
2 said external-access-width control means, as the bit-
3 width information, outputs a bit width determined in accordance
4 with the effective bit-width of the data used in the program
5 when said register information represents the data registers,
6 and outputs a bit-width determined in accordance with a
7 sufficiently large address space for a program size and data
8 area size of the program when said register information

9 represents the address registers.

1 ³¹~~40~~. The processor of Claim ²⁹~~38~~, wherein:

2 the address registers and data registers in said
3 register means are all 24-bit wide;

4 said instruction decoding means outputs information
5 that represents one of the address register and the data
6 register as the register information;

7 said external-access-width control means outputs the
8 bit-width information exhibiting 24 bits when the register
9 information representing the address register, and outputs the
10 bit-width information exhibiting 16 bits when the register
11 information representing the data register; and

12 the external-access executing means executes the data
13 transfer three times and twice for the 24- and 16-bit-width
14 information respectively for an 8-bit-width external memory,
15 and for twice and once for the 24- and 16-bit-width information
16 respectively for a 16-bit-width external memory.

1 ³²~~41~~. The processor of Claim ³¹~~40~~, wherein said access
2 executing means includes:

3 an address generating circuit for holding an address
4 designated by the data-transfer instruction to output one of
5 a byte address and a word address to the external memory;

6 an output data buffer for holding write data designated
7 by the data-transfer instruction to output the same one of per

8 byte and per word to the external memory;
9 an input data buffer for holding data from read out from
10 the external memory; and
11 a sequence circuit for outputting a byte address to said
12 address generating circuit for an 8-bit-width external memory
13 while controlling the number of times for the data-transfer in
14 accordance with the bit-width information via the input/output
15 data buffers with respect to the read/write data, for
16 outputting a word address to said address generating circuit
17 for a 16-bit-width external memory while controlling the number
18 of times for the data-transfer in accordance with the bit-
19 width information via the input/output data buffers with
20 respect to the read/write data.

1 ³³~~42~~. The processor of Claim ²⁹~~38~~, wherein:
2 the address registers and data registers in said
3 register means are all 32-bit wide;
4 said instruction decoding means outputs register
5 information indicating whether the instruction uses the address
6 register or data register;
7 said external-access-width control means outputs the
8 bit-width information exhibiting 24 bits when the register
9 information representing the address register, and outputs the
10 bit-width information exhibiting 16 bits when the register
11 information representing the data register; and
12 the external-access executing means executes the data

13 transfer three times and twice for the 24- and 16-bit-width
14 information respectively for an 8-bit-width external memory,
15 and for twice and once for the 24- and 16-bit-width information
16 respectively for a 16-bit-width external memory.

1 ³⁴~~43~~. The processor of Claim ³³~~42~~, wherein said access
2 executing means includes:

3 an address generating circuit for holding an address
4 designated by the data-transfer instruction to output one of
5 a byte address and a word address to the external memory;

6 an output data buffer for holding write data designated
7 by the data-transfer instruction to output the data one of per
8 byte and per word to the external memory;

9 an input data buffer for holding data read out from the
10 external memory; and

11 a sequence circuit for controlling said address
12 generating circuit to output the byte address for an 8-bit-
13 width external memory while controlling the input and output
14 data buffers to input and output the byte data to transfer the
15 read/write data to the external memory in a matching number
16 of times to the bit-width of the external memory, and for
17 controlling said address generating circuit to output the word
18 address for a 16-bit-width external memory while controlling
19 the input and output data buffers to input and output the word
20 data to transfer the read/write data to the external memory in
21 a matching number of times for the bit-width of the external

22 memory.

1 *Sub A5* 44. A processor for operating certain data in
2 accordance with an instruction in a program, comprising:
3 a first register means for holding N-bit data;
4 a second register means for holding N-bit data,
5 extending means for extending said M-bit data to N bits by
6 copying an MSB of said M-bit data in a direction of an upper
7 order, M being less than N;
8 zero-extending means for extending said M-bit data to
9 N bits by copying a value "0" in a direction of an upper order;
10 operating means for operating an arithmetic operation
11 in accordance with an instruction;
12 instruction control means for decoding an instruction
13 to zero-extend M-bit immediate data when said M-bit immediate
14 data are to be stored in said first register means by the
15 decoded instruction and to sign-extend said M-bit immediate
16 data when said M-bit immediate data are to be stored in said
17 second register means by the decoded instruction, said zero-
18 extended and sign-extended N-bit immediate data being outputted
19 in one of two methods, one method being to send the extended
20 N-bit immediate data from their respective extending means to
21 their respective register means directly, the other being to
22 send the same via the operating means to their respective
23 register means.

1 ³⁴
~~45.~~ The processor of Claim ³⁵~~44~~, wherein
2 said first register means is a group of a plurality of
3 address registers for storing addresses, and
4 said second register means is a group of a plurality of
5 register means for storing data.

1 ³⁷
~~46.~~ The processor of ^{claim}~~45~~, wherein said N is 24 and said
2 M is 16.

1 *Sub A6* ~~47.~~ A processor for operating certain data in
2 accordance with an instruction in a program, comprising:
3 a first register means for holding N-bit data;
4 a second register means for holding N-bit data,
5 sign-extending means for extending said M-bit data to
6 N bits by copying an MSB of said M-bit data in a direction of
7 an upper order, M being less than N;
8 zero-extending means for extending said M-bit data to
9 N bits by copying a value "0" in a direction of an upper order;
10 operating means for operating an arithmetic operation
11 in accordance with an instruction;
12 instruction decoding means for decoding an instruction
13 in the program to detect a first type instruction and a second
14 type instruction, said first type instruction including an
15 instruction to store M-bit immediate data into said first
16 register means, said second type instruction including an

17 instruction to store said M-bit immediate data into said second
18 register means; and
19 control means for outputting said M-bit immediate data
20 to said zero-extending means when the first type instruction
21 is detected, and for outputting said M-bit immediate data to
22 said sign-extending means when the second type instruction is
23 detected, said zero-extended N-bit immediate data and sign-
24 extended N-bit immediate data being outputted in one of two
25 methods, one method being to send the extended N-bit immediate
26 data from their respective extending means to their respective
27 register means directly, the other being to send the same via
28 the operating means to their respective register means.

1 ³⁹~~48~~. The processor of Claim ³⁸~~47~~, wherein
2 said first register means is a group of a plurality of
3 address registers for storing addresses, and
4 said second register means is a group of a plurality of
5 register means for storing data.

1 ⁴⁰~~49~~. The processor of Claim ³⁹~~48~~, wherein
2 said first type instruction includes a data-transfer
3 instruction to store the M-bit immediate data to said first
4 register means, an add instruction to add a value in said first
5 register and the M-bit immediate data, and a subtract
6 instruction to subtract the M-bit immediate data from a value
7 in said first register, and

8 said second type instruction includes a data-transfer
9 instruction to store the M-bit immediate data to said second
10 register means, an add instruction to add a value in said
11 second register and the M-bit immediate data, and a subtract
12 instruction to subtract the M-bit immediate data from a value
13 in said second register.

1 ⁴¹
2 ~~50.~~ The processor of Claim ⁴⁰~~49~~, wherein said N is 24 and
3 said M is 16.

1 ^{Sub A7} 51. A data processing method for executing an
2 instruction that includes an instruction to store M-bit
3 immediate data in an N-bit first register and an N-bit second
4 register, both M and N being integers while M being less than
5 N, said method comprising the steps of:
6 decoding an instruction for selecting one of the first
7 and second register in accordance with a decoded instruction;
8 zero-extending said M-bit immediate data to N bits when
9 said decoded instruction designates the first register, and
10 sign-extending said M-bit immediate data to N bits when said
11 decoded instruction designates the second register; and
12 storing extended N-bit immediate data to the designated
13 register.

1 ⁴³
2 ~~52.~~ The method of Claim ⁴²~~51~~, wherein
3 said first register means is a group of a plurality of

3 address registers for storing addresses, and
4 said second register means is a group of a plurality of
5 register means for storing data.

1 ⁴⁴ 53. The method of Claim ⁴³ ~~52~~, wherein said N is 24 and
2 said M is 16.

1 *Sub A8* 54. A processor for executing a program including an
2 N-bit data arithmetic operation instruction, M-bit and N-bit
3 load/store instruction, M being less than N, a conditional
4 branch instruction, a data-transfer instruction with an
5 external memory, and an instruction having immediate data, said
6 processor comprising:
7 a first register means including a plurality of
8 registers for holding N-bit data;
9 a second register means including a plurality of
10 registers for holding N-bit data;
11 a program counter for holding an N-bit instruction
12 address to output the same to said memory means;
13 fetching means for fetching an instruction from an
14 external memory using the instruction address from said program
15 counter;
16 instruction decoding means for decoding a fetched
17 instruction;
18 executing means for executing all N-bit arithmetic
19 operation instructions and for executing N-bit and M-bit

20 instructions excluding the arithmetic operation instructions,
21 a plurality of flag storing means, each for storing a
22 corresponding flag group changed in response to different bit-
23 widths data in accordance with an execution result of said
24 executing means;
25 flag selecting means for selecting a certain flag group
26 from said plurality of flag storing means in accordance with
27 a conditional branch instruction decoded by said instruction
28 decoding means;
29 branch judging means for judging whether a branching is
30 taken or not with a reference to a flag group selected by said
31 flag selecting means;
32 sign-extending means for extending M-bit data to N bits
33 by copying an MSB of said M-bit data in a higher order;
34 zero-extending means for extending M-bit data to N bits
35 by filling a value "0" in a higher order;
36 compensation instruction control means for compensating
37 contents of said first register means and said second register
38 means using said sign-extending means and said zero-extending
39 means in accordance with a compensation instruction inserted
40 immediately after a machine language instruction for an
41 arithmetic operation that will possibly cause an overflow, said
42 machine language instruction being decoded by said instruction
43 decoding means;
44 external-access-width control means for outputting bit-
45 width information for transmission data in accordance with a

46 type of said register means to which a register indicated by
47 register information belongs, said register information
48 indicating one of said first and second register means;
49 external-access executing means for executing a data
50 transfer between the register and an external memory in
51 accordance with said register information and bit-width
52 information; and
53 immediate control means for outputting M-bit immediate
54 data to said zero-extending means when a decoded instruction
55 includes an instruction to store said M-bit immediate data in
56 said first register means, and for outputting said M-bit
57 immediate data to said sign-extending means when a decoded
58 instruction includes an instruction to store said M-bit in said
59 second register means, said zero-extended and sign-extended
60 immediate data being sent to said first and second register
61 means respectively in two methods, one being to send the same
62 directly to their respective register means and the other being
63 to send the same via said executing means.

- 1 ⁴⁶ 55. The processor of Claim ⁴⁵ 54, wherein said N is 24 and
2 said M is 16.